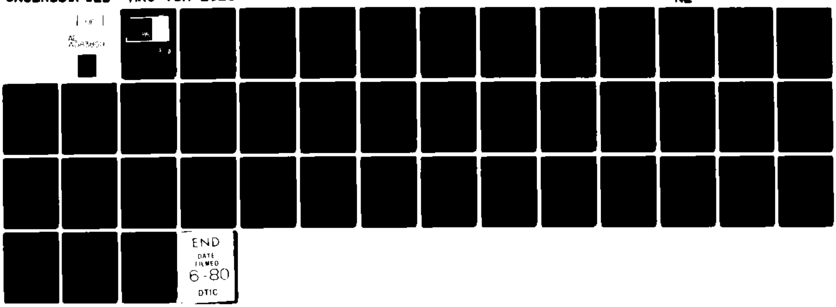
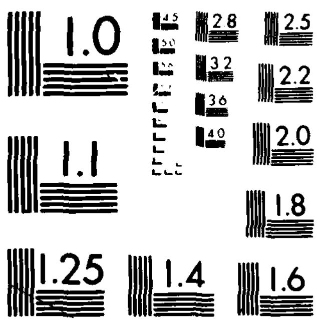
AD-A083 809 WISCONSIN UNIV-MADISON MATHEMATICS RESEARCH CENTER F/G 12/1
COMPUTER CALCULATION OF MECHANISMS INVOLVING INTERMITTENT MOTIO--ETC(U)
DEC 79 B NOBLE, H S HUNG DAAG29-75-C-0024
UNCLASSIFIED MRC-TSR-2026 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

ADA083809

MRC Technical Summary Report #2026

COMPUTER CALCULATION OF MECHANISMS INVOLVING INTERMITTENT MOTIONS

B. Noble [redacted] H. S. Hung

LEVEL II

MRC-TSR-2026

Mathematics Research Center
University of Wisconsin—Madison
610 Walnut Street
Madison, Wisconsin 53706

Dec[redacted]79

43

(Received May 31, 1979)

DAAG29-75-C-0024

Approved for public release
Distribution unlimited

Sponsored by

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park
North Carolina 27709

80 4 9 116

221200

UNIVERSITY OF WISCONSIN-MADISON
MATHEMATICS RESEARCH CENTER

COMPUTER CALCULATION OF MECHANISMS INVOLVING INTERMITTENT MOTIONS

B. Noble and H. S. Hung

Technical Summary Report #2026
December 1979

ABSTRACT

This paper deals with a simple computational approach to the analysis of dynamical systems involving intermittent motion in which the velocities involved can be discontinuous due to impulsive forces, impact, mass capture, and mass release. The sequence of these events may not be known ahead of time, and may in fact be one of the things we wish the computer to determine.

The dynamical equations are formulated using a logical function method due to P. Ehle. The resulting system of ordinary differential equations with discontinuous coefficients is integrated using a standard computer code in regions where the coefficients are continuous. When discontinuities occur, jump conditions across the discontinuity are used to express the new velocities in terms of the old, and the ordinary differential equation solver is simply restarted with new initial conditions.

To illustrate the simplicity of the approach, the method is applied to a dynamical system of ten masses considered by Ehle. The computer code and numerical results are included.

AMS (MOS) Subject Classifications: 65L05, 70.34, 70.65

Key Words: Mechanical systems, Intermittent motion, Heaviside step-functions, Logical functions, Jump conditions, Equations of motion, Dynamical analysis, Computer program

Work Unit Number 3 (Applications of Mathematics)

Sponsored by the United States Army under Contract No. DAAG29-75-C-0024.

SIGNIFICANCE AND EXPLANATION

Apart from the work by Ed Haug and his students at the University of Iowa, surprisingly few references seem to exist on the computer calculation of complicated mechanical systems involving intermittent motion, particularly when the sequence of events is not known beforehand. P. Ehle has formulated such problems using a "logical function" approach involving Heaviside step functions and their derivatives. He then smooths out the discontinuities so that the resulting ordinary differential equations can be integrated directly by a standard computer code. We avoid the somewhat arbitrary choice of smoothing parameters, the calculation of the smoothing functions in the transition regions, and the step-size adjustment through the transition regions, by dealing with the discontinuities directly by using jump conditions across the discontinuities. A computer code is included for an example considered by Ehle, to illustrate the simplicity of the method.

The responsibility for the wording and views expressed in this descriptive summary lies with MRC, and not with the authors of this report.

COMPUTER CALCULATION OF MECHANISMS INVOLVING INTERMITTENT MOTIONS

B. Noble and H. S. Hung

1. Introduction.

There seem to be rather few published articles dealing with computational methods for the analysis of dynamical systems involving intermittent motion in which the velocities involved can be discontinuous due to impulsive forces, impact, mass capture, and mass release. Bickford [1] uses analytic and graphical methods to design such mechanisms, but he does not use computer simulation. The book by Levy and Wilkinson [7] deals with the computer analysis of dynamical systems, including situations in which masses come into contact with elastic 'stops'.

During the last few years extensive work has been carried out by Professor Ed Haug and his students at the University of Iowa in connection with the computer calculation of complicated mechanical systems with intermittent motion. In the earlier work (see, for instance, [4], [5] and [6]) it is assumed that the order of the sequence of events is known a priori. In a complex mechanism, the sequence of events may be highly design dependent, and it may be one of the things that we wish the computer program to discover. P. Ehle [2] has introduced a "logical function" method consisting of two distinct steps to deal with this latter type of situation:

Step 1. The discontinuities are represented in the equations of motion by Heaviside step functions and their derivatives. The arguments of these logical functions can involve space, velocity, or time, whichever is physically appropriate. The motion is represented by one single set of equations over the entire interval of time under consideration.

Step 2. The discontinuities are smoothed out by an ingenious but somewhat arbitrary procedure. The resulting system of ordinary differential equations involves continuous coefficients so that it can be integrated directly by standard computer codes.

Sponsored by the United States Army under Contract No. DAAG29-75-C-0024.

The present paper adopts the logical function approach in Step 1, but, instead of Step 2, deals directly with the resulting system of ordinary differential equations involving discontinuous functions. In regions where the coefficients are smooth, the equations are integrated using a standard computer software code for solving systems of ordinary differential equations. When discontinuities occur, the jump conditions across the discontinuities are used to express the new velocities in terms of the old, and the ordinary differential equation solver is simply started with new initial conditions. The method is easy to implement and avoids the somewhat arbitrary choice of smoothing parameters, the calculation of the smoothing functions in the transition regions, and the step-size adjustment through the transition regions that are required in Ehle's approach.

A simple example is discussed in Section 2 to illustrate the essential points of our approach.

In Section 3 we apply our method to the complicated example considered by Ehle in Chapter 4 of his thesis [2]. The results confirm that our method is easily implemented. In order to facilitate a comparison with Ehle's treatment, we use his notation, and the computer runs are carried out using his numerical parameters, with minor changes noted later. In order to make the present paper self-contained (and also to save the reader the labor of extracting the relevant information from Ehle's thesis), we define in complete detail in Appendices B and C the symbols used in Section 3 below. Our computer code is given in Appendix D.

Most of Ehle's thesis is devoted to sensitivity analysis for the complicated mechanism in Section 3 below. A sensitivity analysis using the method in this paper would be the next natural step in the present work.

2. A Simple Example.

Consider the idealized situation in Figure 1 which will illustrate most of the points required for the analysis of the complicated mechanism in Section 3. Motion is in the x-direction only. The mass A, position $x = x(t)$, is attached to a massless spring with spring constant k_A. The unstressed length of the spring is x_0. When the mass A reaches position $x = x_1$ (for the first time only), an impulsive force of magnitude F acts on it. The mass B is initially at rest. When mass A reaches mass B (at $x = x_2$), the two masses lock together, and move as one. The equations of motion are as follows.

$$m_A\ddot{x} = k_A(x - x_0) + F\delta(t - t_1), \qquad 0 \le x < x_2 , \tag{2.1}$$

where t_1 denotes the (unknown) instant when mass A reaches x_1.

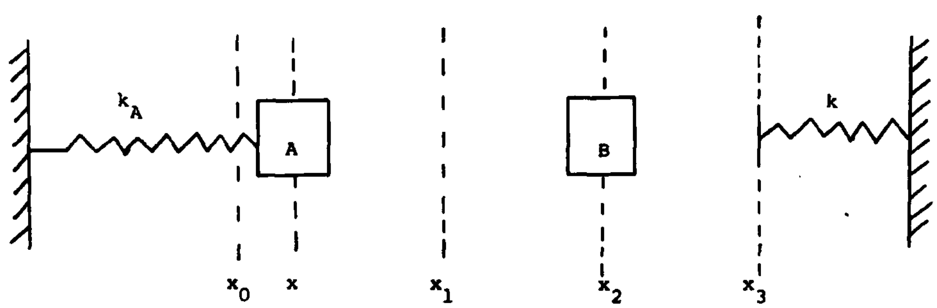


Figure 1. A Simple System

The effect of the impulsive force $F\delta(t - t_1)$ is to produce a jump F/m_A in velocity at $t = t_1$, the position $x(t)$ changing continuously in t at $t = t_1$. This can be seen by integrating (2.1) between $t_1 - \Delta$ and $t_1 + \Delta$, then letting $\Delta \to 0$, which gives

$$m_A \dot{x}(t_1 + 0) - m_A \dot{x}(t_1 - 0) = F \tag{2.2}$$

where we use an obvious notation. A double integration shows similarly that

$$x(t_1 + 0) = x(t_1 - 0) . \tag{2.3}$$

For $x_2 \leq x < x_3$, we have

$$(m_A + m_B)\ddot{x} = k_A(x - x_0) . \tag{2.4}$$

From conservation of momentum as the mass m_A reaches $x = x_2$ and picks up m_B, we see that

$$m_A \dot{x}(t_2 - 0) = (m_A + m_B)\dot{x}(t_2 + 0) , \tag{2.5}$$

and again the displacement is continuous, $x(t_2 - 0) = x(t_2 + 0)$.

For $x \geq x_3$, we have

$$(m_A + m_B)\ddot{x} = k_A(x - x_0) + k(x - x_3) . \tag{2.6}$$

Subsequently (2.4) holds whenever $x < x_3$ and (2.6) whenever $x \geq x_3$.

Following Ehle, we use the Heaviside step-function to write the above three equations as one single equation. We define, for any u,

$$H(u) = \begin{cases} 0, & u < 0 , \\ 1, & u \geq 0 . \end{cases}$$

We also require the following step-function. Suppose that u is some time-dependent quantity such that $u < 0$ for $0 \leq t < T$ and that $u \geq 0$ <u>for the first time</u> at $t = T + \epsilon$ (ϵ arbitrarily small). We then define

$$H_1(u) = \begin{cases} 0, & 0 \leq t < T , \\ 1, & t \geq T , \end{cases}$$

i.e. for $t \geq T$, $H_1(u)$ is always 1 regardless of the size of u.

In terms of these logical step-functions we can rewrite equations (2.1), (2.4), (2.6) as:

$$\frac{d}{dt}\{[m_A + H_1(x - x_2)m_B]\dot{x}\} = k_A(x - x_0) + F\delta(t - t_1) + H(x - x_3)k(x - x_3) \quad . \tag{2.7}$$

Here t_1 is the time when $x = x_1$ <u>for the first time</u>.

As discussed in connection with (2.2), (2.3), the effect of the impulsive form $F\delta(t - t_1)$ is to produce a discontinuity in velocity at $t = t_1$. The effect of the term involving $H_1(x - x_2)$ can be seen by integrating (2.7) between $t = t_2 - \Delta$ and $t = t_2 + \Delta$, where the mass is at $x = x_2$ at time $t = t_2$, then letting $\Delta \to 0$. This gives precisely (2.5). The right-hand side of (2.7) is a continuous function of x, t as x passes through x_3.

Our procedure for solving (2.7) numerically is to use a standard computer code for numerical integration of a system of ordinary differential equations in time intervals in which the mass is not changing and the impulsive force at $t = t_1$ is not acting.

The numerical integration is started with initial conditions $x(0) = 0$ and $\dot{x}(0) = 0$. We check at each step whether $x \geq x_1$. Whenever this condition is satisfied for the first time, we take the impulsive force F into account by restarting the numerical integration with new initial conditions given by (2.2), (2.3). After this point we need not check further whether $x \geq x_1$. (The impulsive force occurs only once.)

To take into account the mass change at $x = x_2$, we similarly check at each step of the numerical integration whether $x \geq x_2$. When this occurs for the first time, we restart the numerical integration with new initial conditions given by (2.5). After this point, masses A and B are locked, $H_1(x - x_2)$ is always 1, and it is no longer necessary to check whether $x \geq x_2$.

Finally, the term $H(x - x_3)k(x - x_3)$ is a continuous function of x and does not require restarting the program with new initial conditions. The term $k(x - x_3)$ is simply added on the right of the equation when $x > x_3$. The additional term is a continuous function of x which is handled directly by the program, i.e., it is not

necessary to restart the differential equation integration as for the other two discontinuities.

In the above example we have assumed that $x_1 > x_2$, i.e., the impulsive force acts before mass B is captured by mass A. If the computer program is arranged so that the conditions "$x \geq x_1$?", "$x \geq x_2$?" are both checked at each time step starting at $t = 0$, and the program is restarted with appropriate initial conditions depending on which condition is satisfied first, then the program will work whether $x_1 < x_2$ or $x_1 > x_2$. This illustrates one of the main points of the method, that it is not necessary to know the sequence of events ahead of time.

Note that the above example discussed the treatment of only some of the possible discontinuous situations due to impulsive forces, impact, mass capture and mass release. For the standard treatment of such discontinuities one could refer to [8], for instance.

3. Computer Calculation of A Complicated Mechanical System.

(a) Description of the mechanism. We shall compute by our method the motion of the mechanism considered by Ehle in Chapter 4 of his thesis [2]. Schematic diagrams are given in Figure 2 which refers to time $t = 0$, and Figure 3 which refers to a later time. Masses 1 - 10 move parallel to the x-axis as indicated; in addition, mass 8 can rotate about an axis parallel to the x-direction, this rotation being controlled by a pin C moving in a slot AB (a cam motion) as shown.

There are only seven equations of motion since at any one time there are only seven independent moving bodies. Bodies $3,\ldots,7$ are simply the masses $m_3,\ldots,m_7$ in Figure 2. Masses 8, 9, 10 are attached to either mass 1 or mass 2 at any given instant of time. We shall use the terminology 'body 1' ('body 2', respectively) to refer to the appropriate combination of m_1, m_8, m_9, m_{10} (m_2, m_8, m_9, m_{10}, respectively) moving as single bodies at a given instant of time. The exact distribution of m_8, m_9, m_{10} between bodies 1 and 2 is controlled by the positions and velocities of bodies 1 and 2 (see equations (3.1)). At time $t = 0$ body 2 has mass m_2, and in Ehle's notation, body 1 has mass $m_1 + 20(m_9 + m_{10})$. In view of the subsequent motion it is convenient to say that at time $t = 0$, body 1 consists of a mass M_1 to which m_8, m_9, m_{10} are attached, i.e., we introduce a new symbol M_1 such that the mass of body 1 at time $t = 0$ is $M_1 + m_8 + m_9 + m_{10}$ (i.e., $M_1 = m_1 + 19(m_9 + m_{10}) - m_8$).

The position of body i in the x-direction is measured by a co-ordinate x_i such that $x_i = i$ at $t = 0$. This co-ordinate system is chosen to ensure that no x_i is ever negative (see [2], p. 66 for more details). The velocity of body i is denoted by $\dot{x}_i$.

Two types of spring-damper pairs are involved in the mechanical system under consideration. Using Ehle's notation, let FIJ denote the force exerted by spring-damper pair J on body I. In one type, the spring-damper pair is connected to bodies I and J and FIJ is proportional to the extension or compression of the

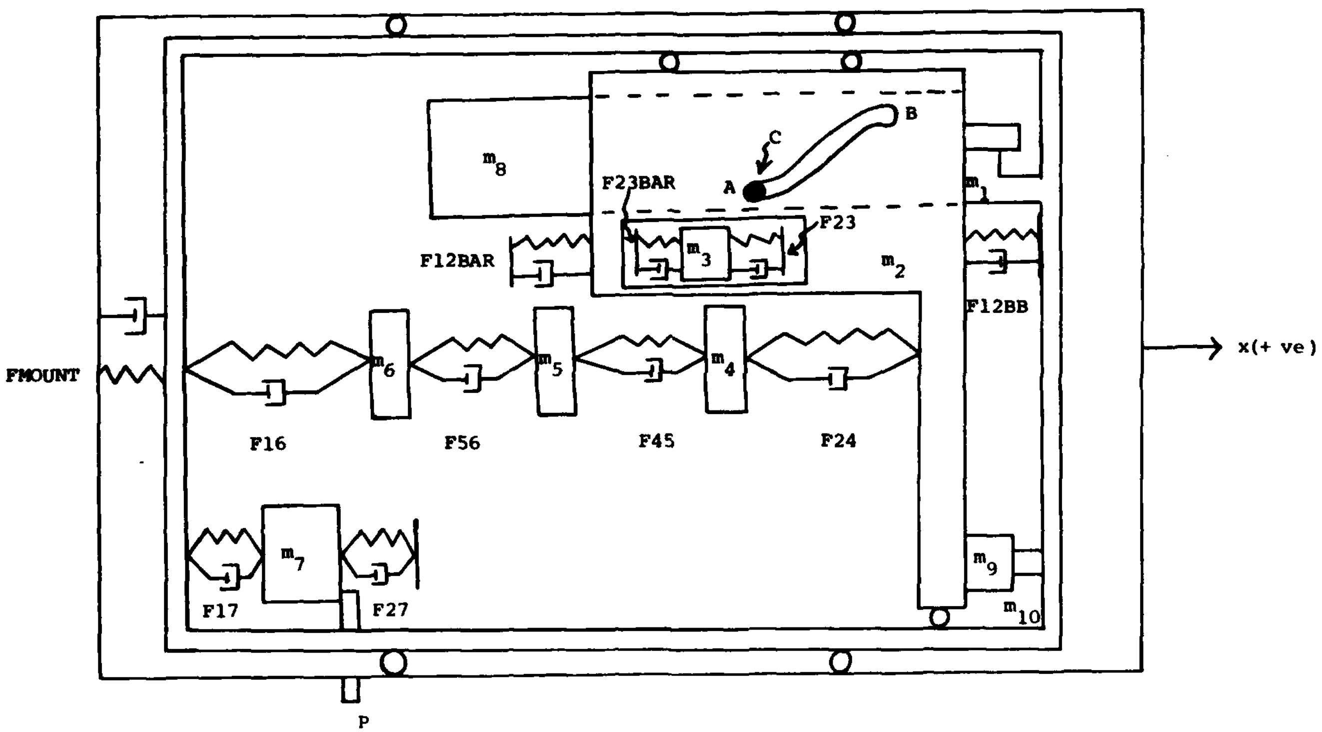


Figure 2. Mechanical System Shown at Time t = 0.

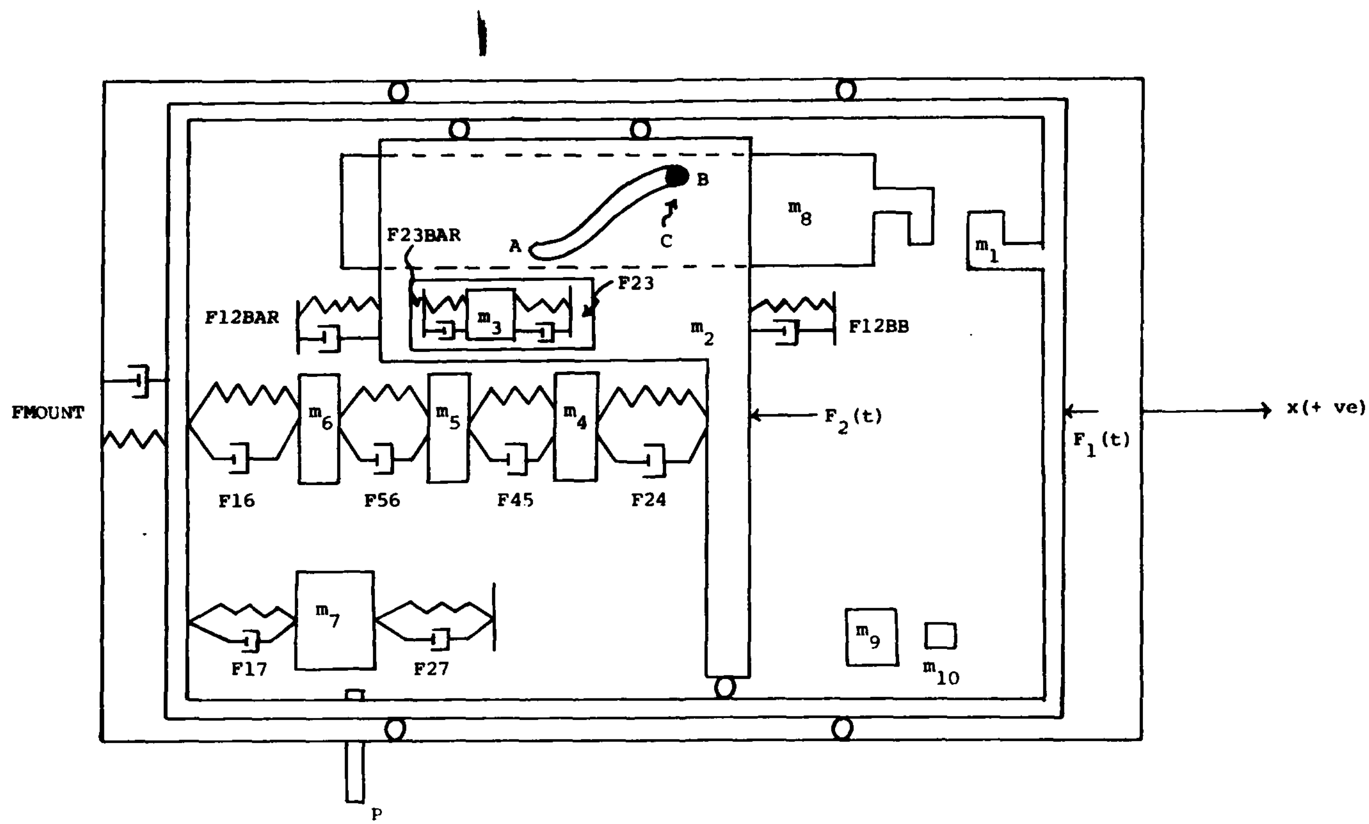


Figure 3. Mechanical System Shown at Later Time.

spring plus a damper force proportional to the velocity difference between the ends (FMOUNT, F16, F17, F24, F45, F56). In the other type, the force is proportional to the compression of the spring plus the damper force. When the distance between the bodies is less than the static or free length of the spring, but when the separation of the bodies exceeds the static length of the spring, the spring loses contact with one of the bodies and FIJ = 0 (F12BAR, F12BB, F23, F23BAR, F27).

At time $t = 0$, body 2 is in its extreme right position (see Figure 2); also F16, F56, F45, F24, F12BB and F17 are in compression, and the corresponding forces balance. The pin P is pulled to release body 7; the spring force F27 is inactive since the spring-damper pair attached to the right side of body 7 is some distance from body 2. The spring force F17 is active; the spring-damper pair between bodies 1 and 7 pushes body 1 to the left and body 7 to the right. Since the mass of body 7 is much less than the mass of body 1, the velocity of body 7 is much greater than that of body 1. Bodies 2, 4, 5, 6 also move due to spring forces. Body 3 is centered in its slot in body 2, and does not move initially.

When the stiff spring on the right side of body 7 strikes body 2, the impulsive forces F_1, F_2 (as shown in Figure 3) act on bodies 1, 2 respectively. At the same time, the mass of body 1 is decreased by m_{10}.

From this point onwards we shall not attempt to describe the motion in words, because it is in fact clearer and simpler to quote the equations used by Ehle in [2] to describe the motion (see (3.1) below).

The objective here is to predict the time histories of the displacements, velocities, and forces associated with each independent rigid body that occurs in one cycle of motion of the mechanism, the end of the cycle being determined when the right spring on mass 2 strikes mass 1.

(b) Equations of motion for the mechanism. We simply quote the following first order differential equations of motion used by Ehle in his thesis [2]:

$$\begin{cases} \dfrac{d}{dt}\,[EMB(I) \times Y(I)] = NF(I) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad , \quad I = 1,\ldots,7 \\ \dfrac{d}{dt}\,Y(I+7) = Y(I) \end{cases} \tag{3.1}$$

with initial conditions:

$$\begin{cases} Y(I) = 0 \\ Y(I+7) = I \end{cases} , \qquad I = 1,\ldots,7 .$$

In equations (3.1),

(1) Y(I), Y(I + 7) are the velocity and position respectively of body I.

$$\begin{cases} Y(I) = \dot{x}(I) \\ Y(I+7) = x(I) \end{cases} , \qquad I = 1,\ldots,7 .$$

(2) EMB(I) is the mass of body I:

$$\begin{cases} EMB(1) = EM(1) + 20 \times (EM(9) + EM(10)) - ELG(1) \times (EM(9) + EM(10)) \\ \qquad + (1 - ELG(7)) \times EM(7) - ELG(2) \times EM(10) + ELG(3) \\ \qquad \times (EM(8) + EM(9) + EM(10)) - ELG(4) \times (EM(8) + EM(9)) , \\ EMB(2) = EM(2) + ELG(9) \times EM(8) + ELG(10) \times EM(9) \\ \qquad + (ELG(11) - ELG(3)) \times (EM(9) + EM(10)) , \\ EMB(I) = EM(I) \quad \text{for} \quad I = 3,\ldots,7 , \end{cases}$$

where EM(I) is simply the mass I with numerical values as follows:

EM(1) = .1925, EM(2) = .0182, EM(3) = .00696, EM(4) = EM(5) = EM(6) = .001383, EM(7) = .002121, EM(8) = .004037, EM(9) = EM(10) = .0004037. (There is a misprint in Ehle's thesis, where EM(2) is given as 0.182.) (Note that we use EM(I) and EMB(I) to distinguish mass I and mass of body I; Ehle use only EM(I) in his thesis to denote mass I. Otherwise we use Ehle's notation.)

(3) NF(I) is the net force on body I:

$$
\begin{cases}
NF(1) = F16 + FMOUNT + ELG(7) \times F17 + ELG(8) \times F12BAR \\
\qquad + ELG(16) \times F12BB - ELG(5) \times FGAS + ELG(6) \times FCAM \\
NF(2) = F24 + ELG(13) \times F23 + ELG(14) \times F23BAR + ELG(15) \times F27 \\
\qquad + ELG(8) \times F21BAR + ELG(16) \times F21BB \\
\qquad - ELG(5) \times (0.9 \times FGAS) - ELG(12) \times FCAM \\
NF(3) = ELG(13) \times F32 + ELG(14) \times F32BAR \\
NF(4) = F42 + F45 \\
NF(5) = F54 + F56 \\
NF(6) = F65 + F61 \\
NF(7) = ELG(7) \times F71 + ELG(15) \times F72
\end{cases}
$$

where

FGAS is the impulsive force on body 1, shown as F_1 in Figure 3,

FCAM is the axial cam force acting between bodies 1 and 2,

FIJ is the force on body I due to the spring-damper pair attached to body J (similarly for FMOUNT, F12BAR, F12BB and F23BAR).

All these forces are described and explained in detail in Appendix B. Note that FGAS and FCAM are not taken into account in exactly the way in which they appear in NF(1) and NF(2) of equations (3.1); they are dealt with by the special but simple method discussed below and as shown in the program in Appendix D.

(4) The ELG(I) appearing in the expressions for EMB(I) and NF(I) are what Ehle called "logical function groups" which are used to switch masses and forces in and out; they are algebraic combinations of Heaviside step-functions EL(I). The definitions of ELG(I) and EL(I) are tabulated and described in detail in Appendix C.

Note that the fourteen equations in (3.1) are for bodies 1-7 only; bodies 8, 9 and 10 do not have separate equations because they do not have separate degrees of freedom - their positions, as mentioned by Ehle, are determined by the positions of bodies 1 and 2. Note also that in Ehle's thesis [2] two equations, in addition to those in (3.1), involving logical step functions and their derivatives, must be

introduced to properly account for two locked-on logical functions of time. For details see [2], p. 74. We avoid this by the use of IF-statements in the computer program.

(c) <u>Numerical solution of the equations of motion</u>. For the numerical solution of the system of equations of motion (3.1), the EPISODE package [3], a variable step and variable order ODE solver, is used. This algorithm will select automatically the appropriate step-size to meet a given criterion. Because of the requirement of this algorithm, the system of equations (3.1) is put into the form:

$$\frac{d}{dt} Y(I) = F(I), \qquad I = 1,\ldots,14 \tag{3.2}$$

where

$$\begin{cases} F(I) = NF(I)/EMB(I) \\ F(I+7) = Y(I) \end{cases}, \qquad I = 1,\ldots,7 \,.$$

These equations will be processed as they stand when FCAM is not active, i.e., when ELG(6) = ELG(12) = 0. Our method of dealing with FCAM is somewhat different from that used by Ehle. FCAM has the form (see Appendix B, where the values of A, B are given):

$$FCAM = A \frac{d}{dt} [Y(2) - Y(1)] + B \,,$$

so that when FCAM is active, i.e., when ELG(6) = ELG(12) = 1, the first two equations in (3.2) have to be solved simultaneously for dY(1)/dt and dY(2)/dt, leading to:

$$\begin{cases} \dfrac{d}{dt} Y(1) = F(1) \\ \dfrac{d}{dt} Y(2) = F(2) \end{cases} \tag{3.3}$$

where

$$F(1) = (A22 \times B1 - A12 \times B2)/D$$

$$F(2) = (A11 \times B2 - A21 \times B1)/D$$

with

$$D = A11 \times A22 - A21 \times A12 \,,$$

$$A11 = EMB(1) + A,\ A12 = -A,\ B1 = NF(1) + B \,,$$

$$A21 = -A,\ A22 = EMB(2) + A,\ B2 = NF(2) - B \,.$$

Equations (3.3) replace the first two equations of (3.2) when FCAM is active.

Our procedure is to use EPISODE to solve (3.2) in time intervals in which the mass is not changing, and the impulsive forces (F_1 = FGAS, and F_2 = 0.9 × FGAS) are not acting. But whenever a mass changes or the impulsive forces occur, the jump conditions across the discontinuities (deduced as for the simple example in Section 2) are used to express the new velocities in terms of the old and the ODE solver is simply restarted with new initial conditions.

We integrate (3.2) for one cycle of motion with the EPISODE options:

(1) The variable step, variable order implicit Adams method in combination with the functional (fixed point) iteration method,

(2) Relative error control,

(3) Relative error of 1 in 10^{-5}.

These are indicated by MF = 10, IERROR = 3 and EPS = 1.0D-5, respectively in the program in Appendix D.

The essence of our numerical procedure can be found in the flow chart in Figure 4. (Refinements like the impulsive and cam forces are easily added.) Note that we check whether the masses change, and print out results, at steps of Δt in time, where Δt is chosen by the programmer. During any one of these steps, the differential equation solver automatically adjusts the step-size that it has to use to obtain the required accuracy, and these step-sizes may be considerably less than Δt.

The entire computer program (apart from EPISODE) is given in Appendix D.

(d) Discussion of numerical results. Figures 5-10 below show the graphs presented by Ehle in [2] for motion of the mechanism described above, together with the graphs obtained by the method of this paper (dotted lines). We have used exactly the same equations and constants as Ehle with one exception, namely the force terms F23 = -F32 and F23BAR = -F32BAR, as discussed in the next paragraph and in Appendix B.

With one exception, the results are in general agreement, including the sequence of events and the chattering of the spring 27 with body 2. The exception is the motion

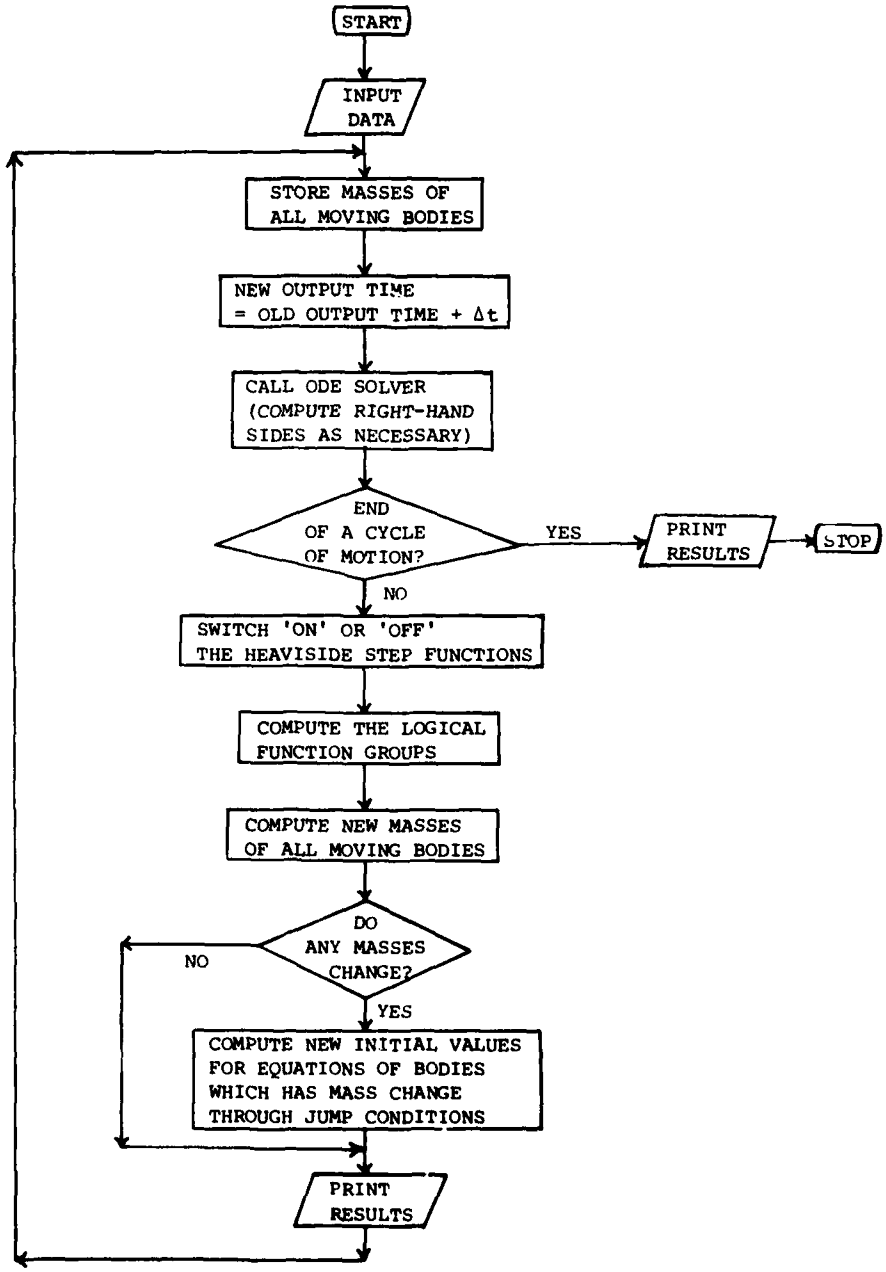
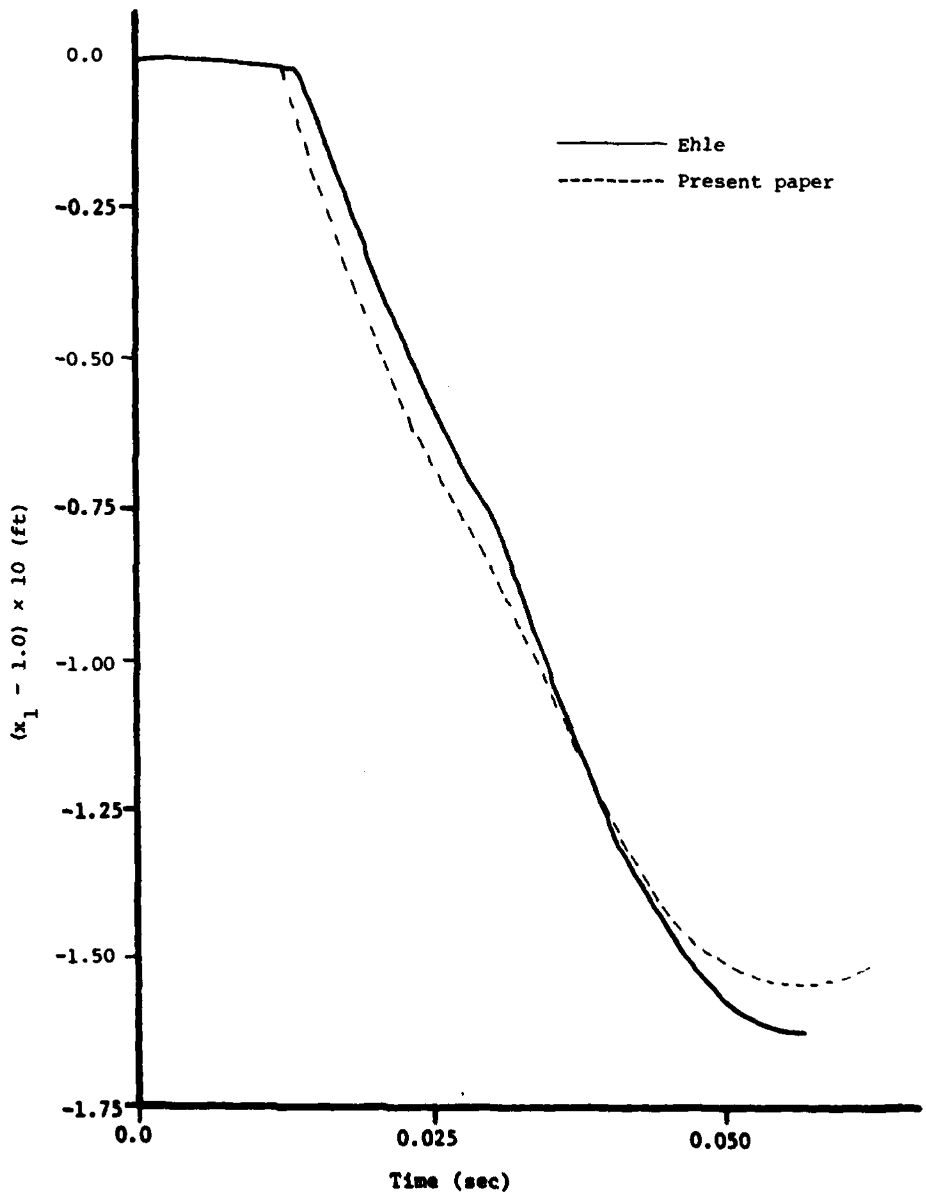
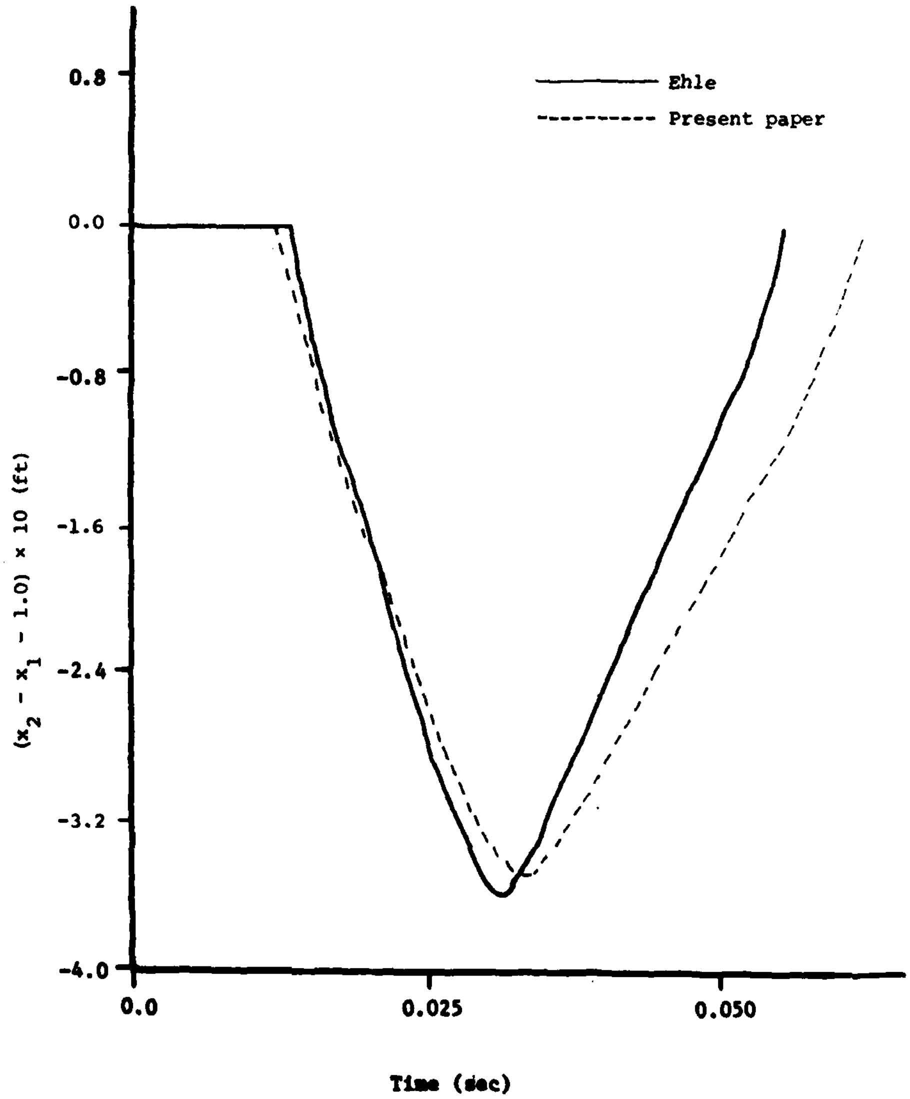


Figure 4. A Generalized Flowchart for Intermittent Motion Problems using Logical Step-Functions

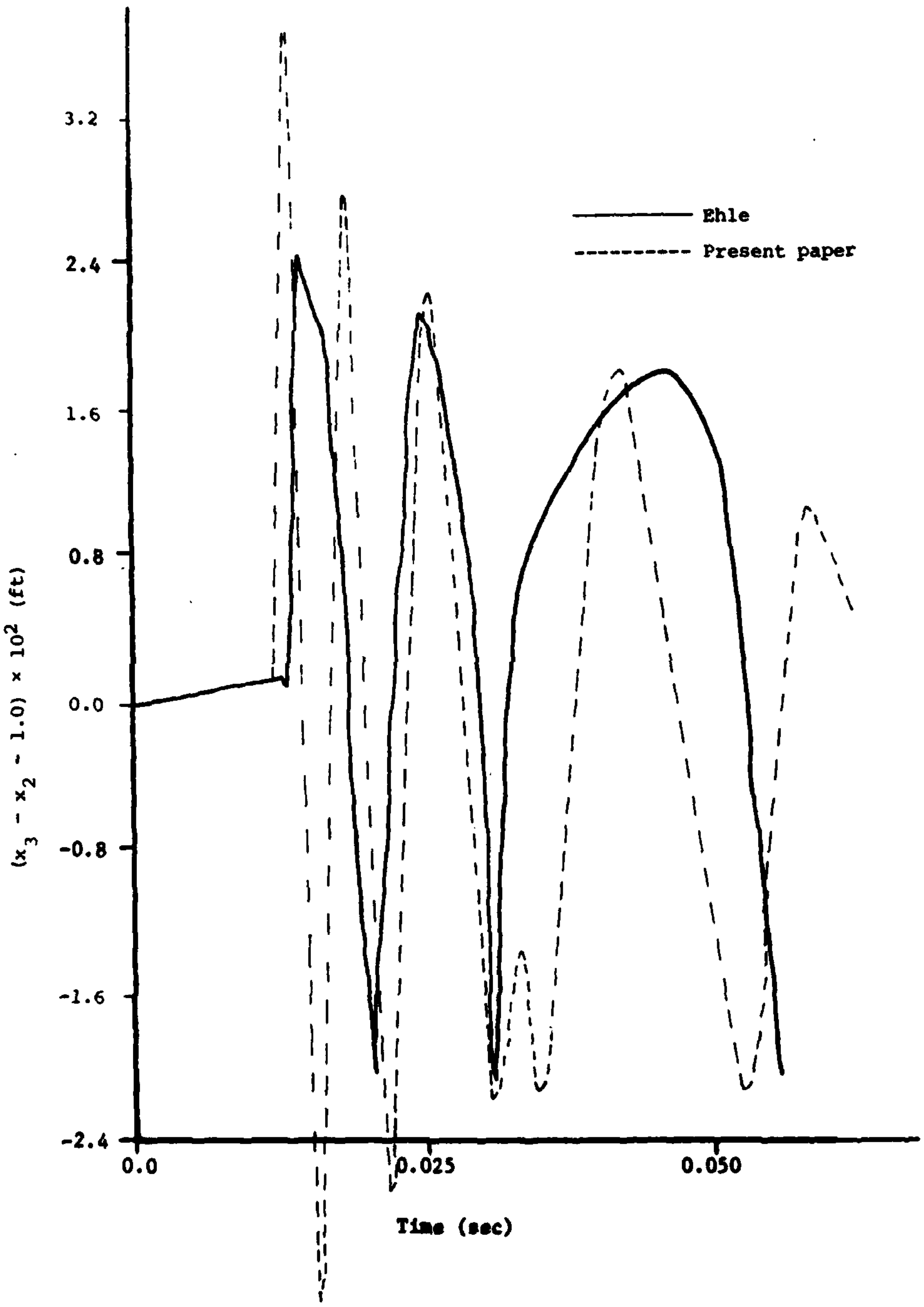


(Ehle: Figure 4.9-15 Body 1 Displacement History)

Figure 5. Displacement of Body 1.

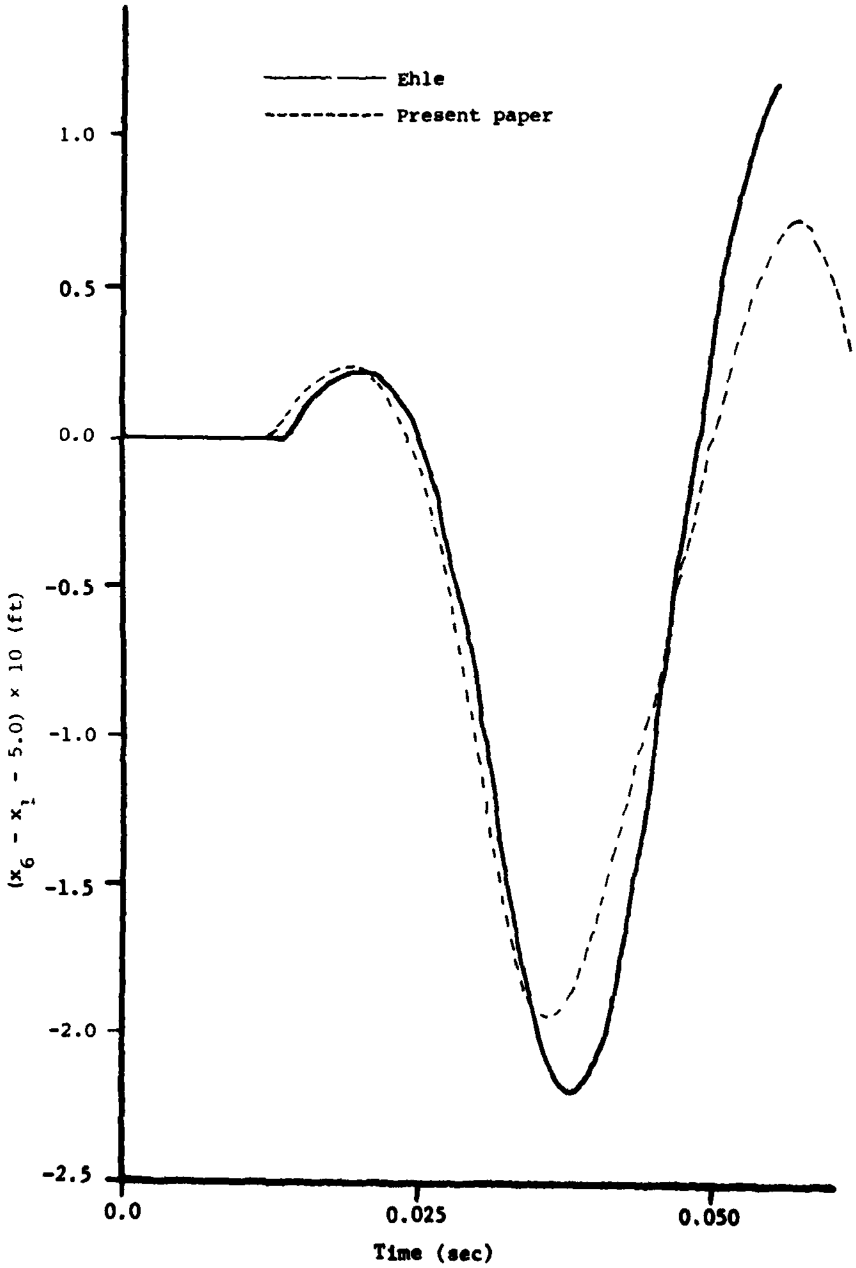


(Ehle: Figure 4.9-16 Body 2 Displacement History)
Figure 6. Difference of Displacements of Bodies 2 and 1.



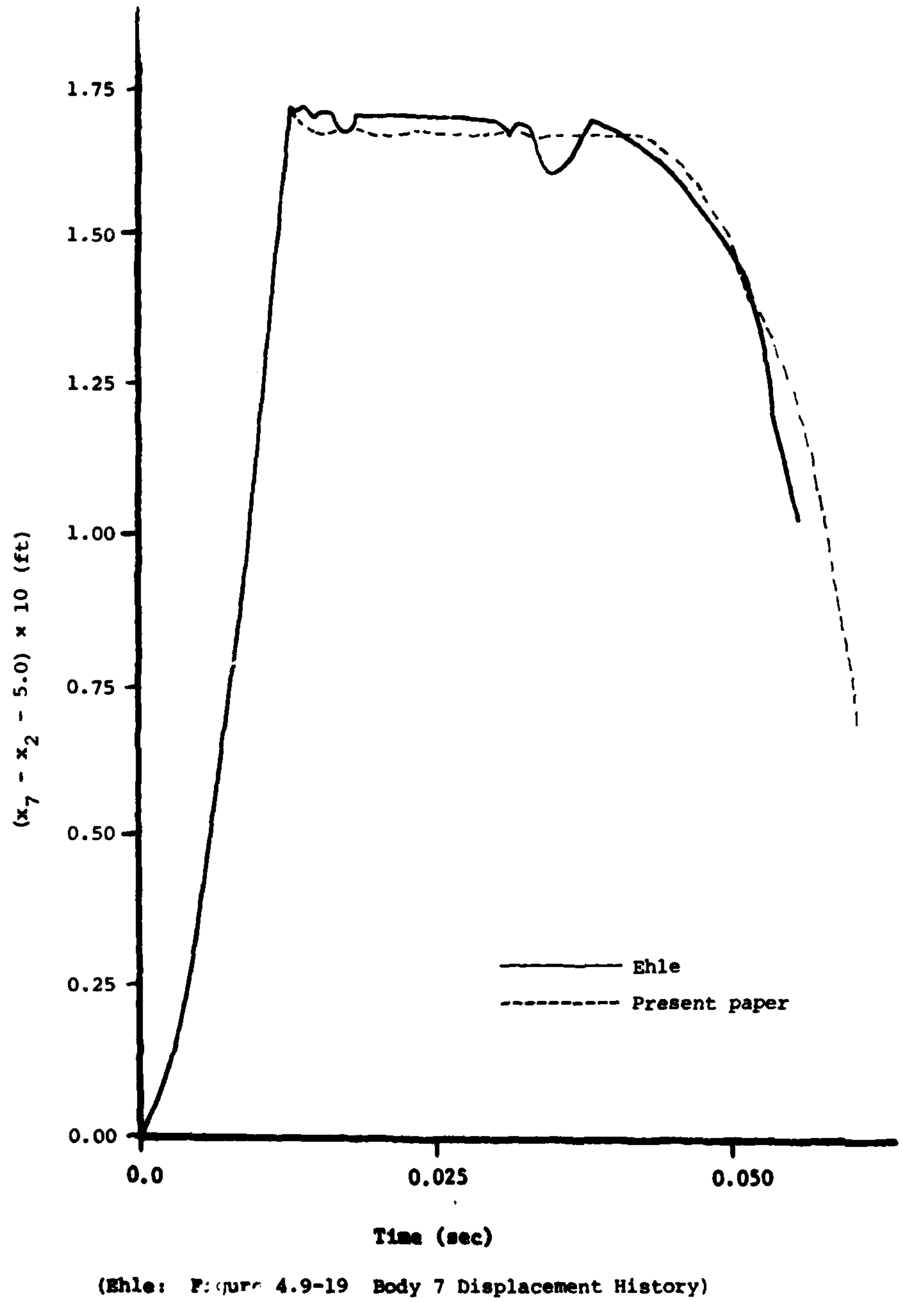
(Ehle: Figure 4.9-17 Body 3 Displacement History)

Figure 7. Difference of Displacements of Bodies 3 and 2.



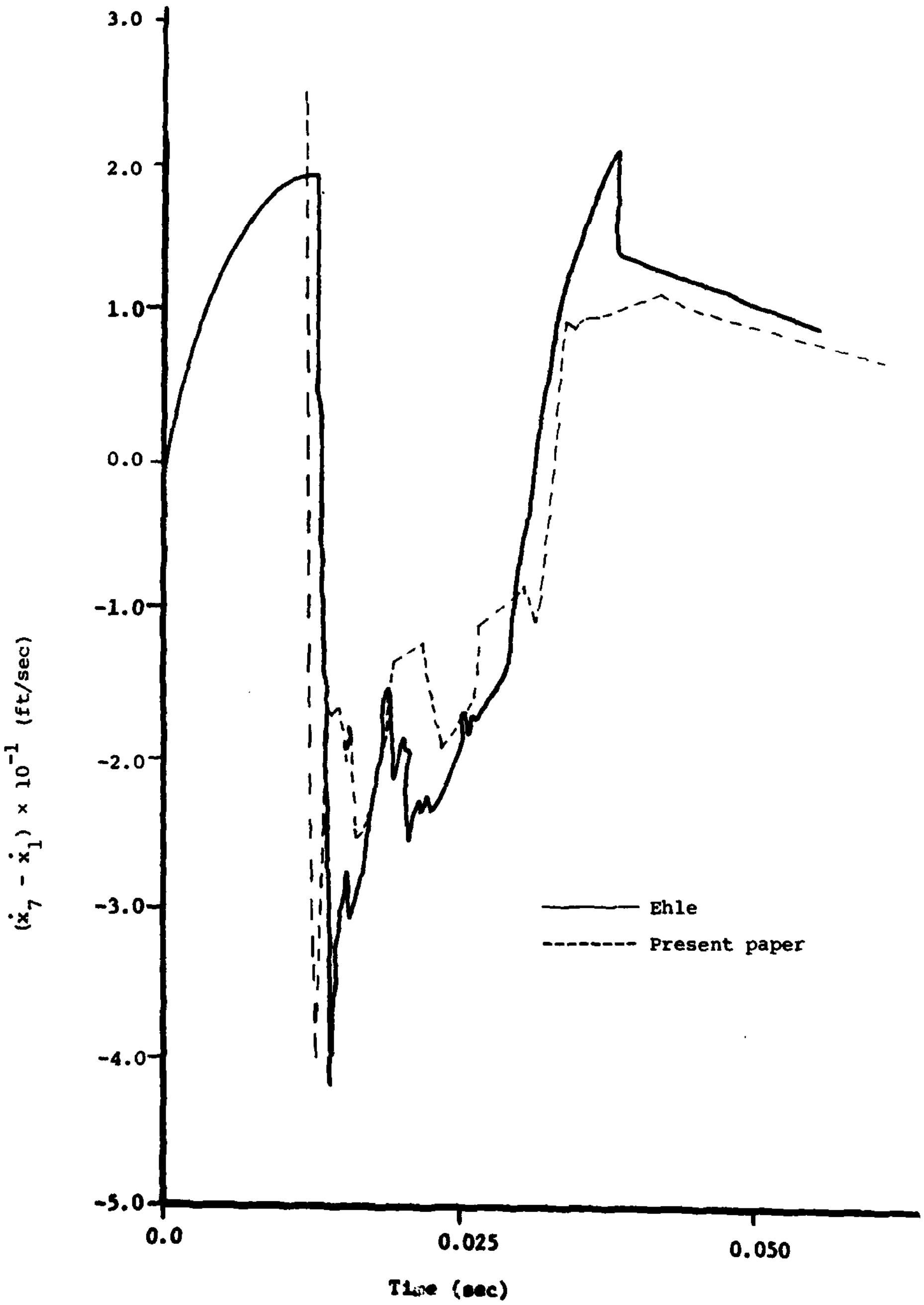
(Ehle: Figure 4.9-18 Body 6 Displacement History)

Figure 8. Difference of Displacements of Bodies 6 and 1.



(Ehle: Figure 4.9-19 Body 7 Displacement History)

Figure 9. Difference of Displacements of Bodies 7 and 2.



(Ehle: Figure 4.9-20 Body 7 Velocity History)

Figure 10. Difference of Velocities of Bodies 7 and 1.

of body 3, the springs attached to which are in intermittent contact with body 2. The value 60 of the damping constant in F23 and F23BAR used by Ehle produced a highly damped motion of body 3 in our program. Our results given in Figure 7 were obtained using a damping constant of 2.

The sequence of events is shown in detail in Figure 11, which describes 32 intermittent contacts at times $t_1, \ldots, t_{32}$. These times are tabulated in Figure 12 below. The criteria involved at each of these times can be found in the definitions of the corresponding logical groups which are described in detail in Appendix C.

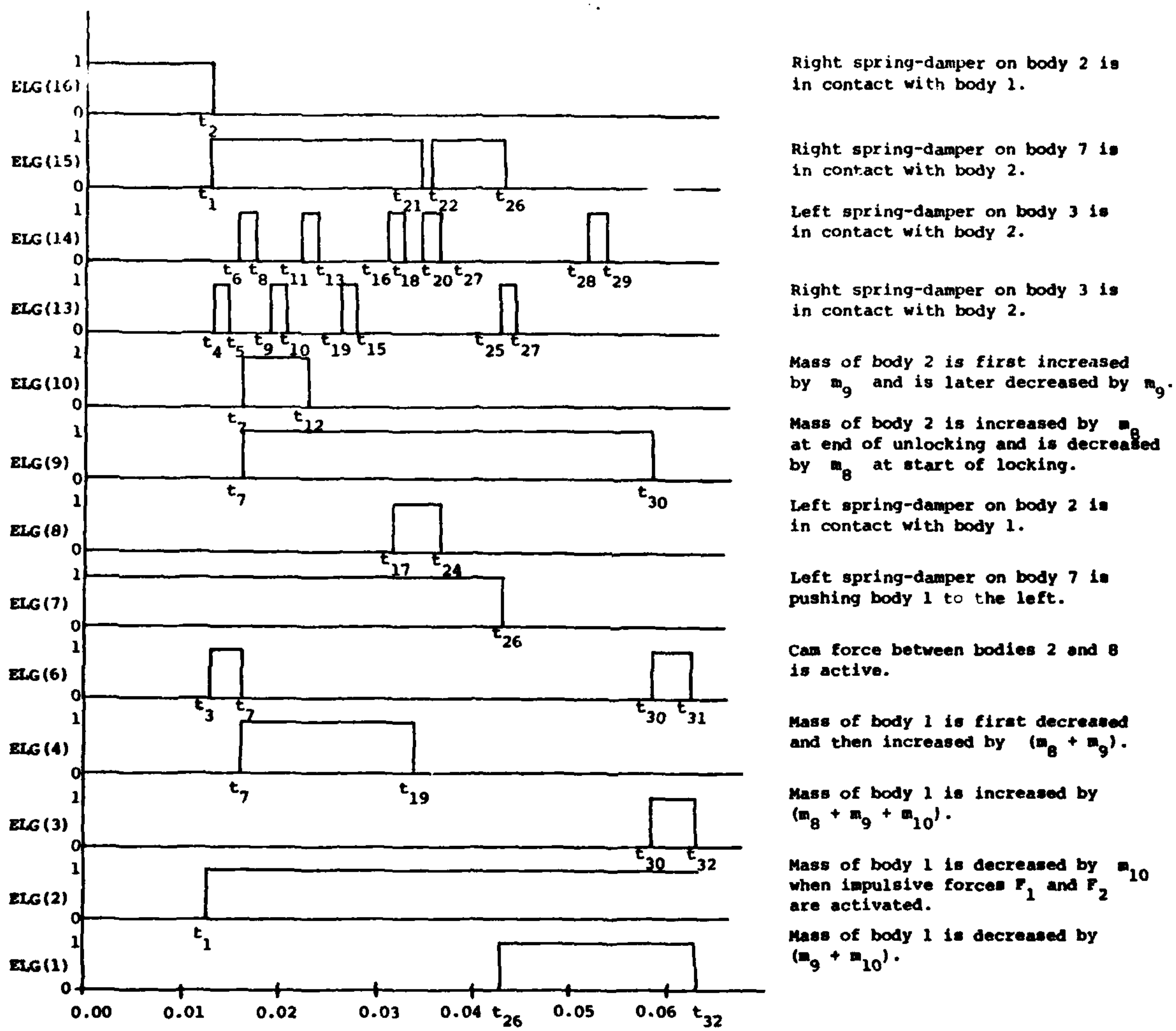


Figure 11. The Sequence of Logical Events

i	t_i
0	0.0000 (Initial time)
1	0.0126
2	0.0127
3	0.0128
4	0.0130
5	0.0146
6	0.0156
7	0.0161
8	0.0173
9	0.0187
10	0.0203
11	0.0221
12	0.0227
13	0.0237
14	0.0262
15	0.0278
16	0.0308
17	0.0316
18	0.0326
19	0.0338
20	0.0344
21	0.0345
22	0.0353
23	0.0364
24	0.0365
25	0.0425
26	0.0429
27	0.0441
28	0.0515
29	0.0533
30	0.0583
31	0.0625
32	0.0629 (End of a cycle of motion)

Figure 12. Times of the Logical Events

4. Summary and Concluding Remarks.

A straightforward method has been presented for the dynamic analysis of mechanisms involving intermittent motion.

1. By using the Heaviside step functions, complicated logic associated with discontinuities in the equations of motion is incorporated systematically into the problem formulation, following the methods introduced by Ehle, but differing from them in detail.
2. Jump conditions are required to get across the discontinuities but these are easily implemented in the program.
3. No a priori knowledge of the order of logical events is required.
4. Our method is easy to program. The program itself is simple; most of the program is just computing the forces and switching Heaviside step functions 'on' or 'off'.
5. The validity of the method has been demonstrated for a complicated and realistic 10-mass mechanical system, the solution involving 32 intermittent contacts (see Figure 11).
6. Computational efficiency of the method is good. We integrate the system of equations of motion by using EPISODE (the variable step and variable order ODE solver) with 5-place accuracy and relative error control from $t = 0$ to $t = 0.063$ which is one cycle of motion of the mechanism, the cpu time on a UNIVAC 1110 computer is 36 seconds. (Note that our version of EPISODE uses double precision. Single precision should suffice, which would reduce the time required.)
7. Stability of the numerical solution does not seem to be a problem.
8. The point of the present paper is that we have adopted the logical function approach of Ehle for dealing with discontinuous motion, but we have dealt with discontinuities directly via jump conditions, instead of smoothing out the discontinuities as done by Ehle [2]. A table of comparison between Ehle's approach and ours is presented in Appendix A.

REFERENCES

[1] J. H. Bickford, "Mechanisms for intermittent motion", Industrial Press, Inc., New York, NY, 1972.

[2] P. Ehle, "Dynamic analysis and design sensitivity analysis of mechanisms with intermittent motion", Ph.D. Dissertation, The University of Iowa, Iowa City, IA, 1978.

[3] A. C. Hindmarsh and G. D. Byrne, "EPISODE: An experimental package for integration of systems of ordinary differential equations", Lawrence Livermore Laboratory, Rept. UCID-30112 (1975).

[4] R.-C. Huang, "Optimal design of mechanical systems with intermittent motion", Ph.D. Dissertation, The University of Iowa, Iowa City, IA, 1976.

[5] R.-C. Huang and E. J. Haug, "A distribution theoretic approach to the sensitivity analysis of mechanisms with intermittent motion", Technical Report No. 42, Material Division, College of Engineering, The University of Iowa, Iowa City, IA, 1978.

[6] R.-C. Huang, E. J. Haug and J. G. Andrews, "Sensitivity analysis and optimal design of a mechanical system with intermittent motion", ASME Journal of Mechanical Design 100 (1978), pp. 492-498.

[7] S. Levy and J. P. D. Wilkinson, "The Component Element Method in Dynamics", McGraw Hill International Book Co., 1976.

[8] A. Seireg, "Mechanical systems analysis", Scranton, PA., International Textbook Co., 1969.

APPENDIX A

TABLE OF COMPARISON BETWEEN EHLE'S APPROACH AND OUR APPROACH

	Ehle's Approach	Our Approach
1.	Use Heaviside step and Dirac delta functions systematically in the formulation of equations of motion.	We follow Ehle in this respect, but write equations in such a way that we use only Heaviside step functions (no Dirac delta functions) except for the impulsive force.
2.	Discontinuities are smoothed out by using "logical functions*" (i.e., smooth approximations to Heaviside step functions and their derivatives).	Discontinuities are dealt with directly by using straightforward jump conditions.
3.	Locked-on logical functions are dealt with via additional ordinary differential equations.	Locked-on logical functions are dealt with via simple computer logic.
4.	To put equations of motion into standard form required by the ODE solver, Dirac delta functions would appear.	This does not occur in our method.
5.	To deal with the logical functions, the transition zone width ϵ is chosen arbitrarily; extra time steps must be taken through the transition zone to calculate the logical functions which involves additional step-size adjustment.	Dealing with the Heaviside step functions is straightforward -- we simply switch them 'on' or 'off' at each time step.

6.	Not so efficient computationally.	Efficient computationally.
7.	Program seems complicated.	Program simple (see Appendix D).

*<u>Note on Ehle's smoothed Heaviside function</u>:

If u is the argument, the smooth representation EL used by Ehle in [2] for the Heaviside step function in the equations of motion is:

$$EL = H(u) = \frac{\frac{1}{2}\left(|u|^3 + u^3\right)}{|u|^3 + \left[|u-\epsilon|^3 - (u-\epsilon)^3\right]}$$

where ϵ is the precise width of the transition zone. Note that the representation EL is asymmetrical about $u = \frac{1}{2}\epsilon$. Symmetry is desirable, and this is easily accomplished by multiplying the square bracket in the denominator of EL by a factor of 0.5.

APPENDIX B

DEFINITIONS OF FORCE TERMS IN EQUATIONS OF MOTION

The forces FGAS and FCAM presented in the equations of motion (3.1) are defined as follows:

FGAS = $1.2 \times \delta(t - t^*)$, where t^* is the time at which the impulsive forces F_1 and F_2 shown in Figure 3 are initiated.

$$\text{FCAM} = A \times \frac{d}{dt}[Y(2) - Y(1)] + B, \text{ where}$$

$$A = \text{GAMLOC}^2 \times \text{ENRT} \times \text{ELP}^2 ,$$

$$B = -\text{GAMLOC}^2 \times \text{ENRT} \times \text{ELP} \times \text{ELPP} \times (Y(2) - Y(1))^2,$$

with

$$\text{GAMLOC} = 0.3927,$$

$$\text{ENRT} = 10.5 \times 10^{-7} + \text{ELG}(1) \times 10^{-7} ,$$

$$\text{ELP} = \frac{1}{2}\left(\frac{\pi}{\text{EPSLOC}}\right) \text{SIN}\left(\frac{\pi}{\text{EPSLOC}}[Y(8) - Y(9) + 0.99167]\right) ,$$

$$\text{ELPP} = \frac{1}{2}\left(\frac{\pi}{\text{EPSLOC}}\right)^2 \text{COS}\left(\frac{\pi}{\text{EPSLOC}}[Y(8) - Y(9) + 0.99167]\right) ,$$

$$\text{EPSLOC} = 0.075.$$

(For details see Appendix D in [2].)

We next tabulate the various spring and damper forces. The equation of motion of a single mass is simply $m\ddot{x} + c\dot{x} + kx = f$. The damping constants c for the mechanism considered are less than the critical damping value, i.e., $c^2 < 4km$.

FMOUNT = -300 × (Y(8) - 1) - 9.53 × Y(1)

Force from spring-damper pair acting between ground and left hand side of body 1.

F12BAR = -10000 × (Y(8) - Y(9) + 0.66667) - 0.13 × (Y(1) - Y(2))

Force on body 1 from spring-damper pair attached to the left side of body 2.

F12BB = -20000 × (Y(8) - Y(9) + 0.99967648) - 25 × (Y(1) - Y(2))

Force on body 1 from spring-damper pair attached to the right side of body 2.

F16 = -76.8 × (Y(8) - Y(13) + 5.08425) - (0.0217 × SQRT(2 × EM(5) × 76.8)) × (Y(1) - Y(6))

Force on body 1 from spring-damper pair between bodies 1 and 6.

F17 = -15 × (Y(8) - Y(14) + 6.5) - (0.98 × SQRT(2 × EM(7) × 15)) × (Y(1) - Y(7))

Force on body 1 from spring-damper pair between bodies 1 and 7.

F23 = -20000 × (Y(9) - Y(10) + 1.016667) - 2 × (Y(2) - Y(3))

Force on body 2 from spring-damper pair attached to right side of body 3.

F23BAR = -20000 × (Y(9) - Y(10) + 0.98333) - 2 × (Y(2) - Y(3))

Force on body 2 from spring-damper pair attached to left side of body 3.

F24 = -76.8 × (Y(9) - Y(11) + 1.91575) - (0.0217 × SQRT(2 × EM(5) × 76.8)) × (Y(2) - Y(4))

Force on body 2 from spring-damper pair between bodies 2 and 4.

F27 = -20000 × (Y(9) - Y(14) + 5.16667) - 36 × (Y(2) - y(7))

Force on body 2 from spring-damper pair attached to right side of body 7.

F45 = -76.8 × (Y(11) - Y(12) + 0.91575) - (0.0217 × SQRT(2 × EM(5) × 76.8)) × (Y(4) - Y(5))

Force on body 4 from spring-damper pair between bodies 4 and 5.

F56 = -76.8 × (Y(12) - Y(13) + 0.91575) - (0.0217 × SQRT(2 × EM(5) × 76.8)) × (Y(5) - Y(6))

Force on body 5 from spring-damper pair between bodies 5 and 6.

F21BAR = -F12BAR

F21BB = -F12BB

F32 = -F23

F32BAR = -F23BAR

F42 = -F24

F54 = -F45

F61 = -F16

F65 = -F56

F71 = -F17

F72 = -F27

(Note that in F23 and F23BAR we use a damper constant of 2 instead of 60 used by Ehle. Also Ehle uses, instead of the above F23BAR,

F23BAR = 20000 × (Y(9) - Y(10) - 0.98333) + 60 × (Y(2) - Y(3)).)

APPENDIX C

DEFINITIONS OF LOGICAL GROUPS AND LOGICAL STEP FUNCTIONS IN EQUATIONS OF MOTION

The logical groups ELG(I) that appear in equations (3.1) are expressed in terms of logical step functions EL(I) and are interpreted in terms of physical events as follows:

ELG(1) = EL(1) × EL(4) × (1 - EL(15))

Mass of body 1 is decreased by EM(9) + EM(10).

ELG(2) = EL(2)

Mass of body 1 is decreased by EM(10).

ELG(3) = EL(3) × ELG(1)

Mass of body 1 is increased by EM(8) + EM(9) + EM(10).

ELG(4) = EL(19) × EL(20)

Mass of body 1 is first decreased and then increased by EM(8) + EM(9).

ELG(5) = ELG(2)

ELG(6) = EL(7) × EL(8) - EL(9) × EL(10) + EL(11) × EL(12) - EL(13) × EL(14)

Cam force between bodies 2 and 8 is active.

ELG(7) = 1 - ELG(1)

Left spring-damper on body 7 is pushing body 1 to the left.

ELG(8) = EL(18)

Left spring-damper on body 2 is in contact with body 1.

ELG(9) = EL(19)

Mass of body 2 is increased by EM(8) at end of unlocking and is decreased by EM(8) at start of locking.

ELG(10) = EL(20) × (EL(19) - EL(21))

Mass of body 2 is first increased by EM(9) and is later decreased by EM(9).

ELG(11) = ELG(1)

Mass of body 2 is increased by EM(9) + EM(10).

ELG(12) = ELG(6)

ELG(13) = EL(22)

Right spring-damper on body 3 is in contact with body 2.

ELG(14) = EL(23)

Left spring-damper on body 3 is in contact with body 2.

ELG(15) = EL(24) × ELG(7)

Right spring-damper on body 7 is in contact with body 2.

ELG(16) = EL(15)

Right spring-damper on body 2 is in contact with body 1.

The logical step-functions that form the logical groups, their arguments, and their physical event associations are given below. Note that

$$\Delta(Y(I) - Y(J)) \equiv (Y(I) - Y(J))\big|_{t} - (Y(I) - Y(J))\big|_{t=0} .$$

EL(1) = H(Y(9) - Y(8) - 0.7292)

Mass of body 1 is decreased by EM(9) + EM(10) when Δ(Y(8) - Y(9)) = 3.25".

EL(2) = H(Y(14) - Y(9) - 5.16667)

Mass of body 1 is decreased by EM(10) when impulsive forces F_1 and F_2 are activated where Δ(Y(14) - Y(9)) = 2".

EL(3) = H(Y(9) - Y(8) - 0.91667)

Body 8 contacts body 1 for start of locking when Δ(Y(8) - Y(9)) = 1".

EL(4) = H(Y(2) - Y(1))

Body 8 contacts body 1 for start of locking when Y(2) - Y(1) > 0.

EL(7) = EL(3)

EL(8) = EL(4)

EL(9) = H(Y(9) - Y(8) - 0.99167)

Locking stops when Δ(Y(8) - Y(9)) = 0.1".

EL(10) = H(Y(2) - Y(1)) = EL(8)

Locking stops when Y(2) - Y(1) > 0.

EL(11) = H(Y(8) - Y(9) + 0.99167) = 1 - EL(9)

Unlocking begins when Δ(Y(8) - Y(9)) = 0.1".

EL(12) = H(Y(1) - Y(2)) = 1 - EL(8)

Unlocking begins when Y(2) - Y(1) < 0.

EL(13) = H(Y(8) - Y(9) + 0.91667) = 1 - EL(3)

Unlocking ends when Δ(Y(8) - Y(9)) = 1".

EL(14) = H(Y(1) - Y(2)) = EL(12)

Unlocking ends when Y(2) - Y(1) < 0.

EL(15) = H(Y(9) - Y(8) - 0.99899)

Right spring on body 2 is in contact with body 1 when Δ(Y(8) - Y(9)) > 0.

EL(18) = H(Y(8) - Y(9) + 0.66667)

Left spring on body 2 begins contact with body 1 when Δ(Y(8) - Y(9)) = 4".

EL(19) = H(Y(8) - Y(9) + 0.91667) = EL(13)

Mass of body 2 is increased by EM(8) + EM(9) + EM(10) when Δ(Y(8) - Y(9)) = 1".

EL(20) = H(Y(1) - Y(2))

Body 9 remains attached to body 1 until Y(2) - Y(1) < 0.

EL(21) = H(Y(8) - Y(9) + 0.79167)

Body 8 is decreased by EM(9) when Δ(Y(9) - Y(8)) = 2.5".

EL(22) = H(Y(10) - Y(9) - 1.01667)

Right spring-damper on body 3 contacts body 2 when Δ(Y(9) - Y(10)) = 0.2".

EL(23) = H(Y(9) - Y(10) + 0.98333)

Left spring-damper on body 3 contacts body 2 when Δ(Y(9) - Y(10)) = 0.2".

EL(24) = H(Y(14) - Y(9) - 5.16667) = EL(2)

Right spring-damper on body 7 contacts body 2 when Δ(Y(14) - Y(9)) = 2".

APPENDIX D

COMPUTER PROGRAM FOR THE CALCULATION OF THE TEN-MASS MECHANICAL SYSTEM

```
C A PROGRAM FOR THE DYNAMIC ANALYSIS OF AN INTERMITTENT MOTION MECHANISM
C     (1) IT IS A TEN-MASS SYSTEM
C     (2) THE ORDER OF THE SEQUENCE OF EVENTS IS NOT KNOWN A PRIORI
C
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      INTEGER EL,ELG
      DIMENSION Y0(14),EL(24),ELG(16),EM(10),EMB(10),OLDEMB(10)
      COMMON EL,ELG,KEL2,EM,EMB
C INPUT DATA
C   MASSES EM(I):
      DATA EM/0.1925D0,0.0182D0,0.00696D0,3*0.001383D0,0.002121D0,
     *0.0040370D0,2*0.0004037D0/
C   INITIAL VELOCITIES (Y0(I),I=1,7) AND INITIAL DISPLACEMENTS
C   (Y0(I),I=7,14):
      DATA Y0/7*0.0D0,1.0D0,2.0D0,3.0D0,4.0D0,5.0D0,6.0D0,7.0D0/
C   INITIAL RELATIVE VELOCITIES AND RELATIVE DISPLACEMENTS OF INTEREST:
      DATA DX2X1,DX3X2,DX7X2,DV7V1,DX6X1,X1M1/6*0.0D0/
C   INITIAL VALUES FOR LOGICAL FUNCTION GROUPS ELG(I):
      DATA ELG /6*0,1,8*0,1/
C   INITIAL VALUE FOR KEL2, A COUNTER TO COUNT THE NUMBER OF TIMES
C   EL(2) SWITCHES ON:
      DATA KEL2/0/
C   TOTAL TIME ALLOWED,TTL,AND TIME STEP FOR PRINTING OUTPUT,TSTEP:
      DATA TTL,TSTEP/0.10D0,0.0001D0/
C   VALUES OF N,T0,TOUT,EPS,IERROR,MF FOR THE DRIVER SUBROUTINE DRIVE OF
C   EPISODE, THE O.D.E. SOLVER USED IN THIS PROGRAM:
      DATA N,T0,TOUT,EPS,IERROR,MF/14,2*0.0D0,1.0D-5,3,10/
C
C PRINT HEADING AND OUTPUT RESULTS AT INITIAL TIME
      WRITE(6,201) T0,DX2X1,DX3X2,DX7X2,DV7V1,DX6X1,X1M1
  201 FORMAT(1H1,19X,'T',10X,'10(X2-X1-1)',4X,'100(X3-X2-1)',5X,'10(X7-X
     *2-5)',5X,'0.1(V7-V1)',6X,'10(X6-X1-5)',6X,'10(X1-1)',//15X,D10.3,
     *6D16.5)
C COMPUTE MASS OF BODY I, EMB(I), AT INITIAL TIME
      EMB(1)=EM(1)+20.0D0*(EM(9)+EM(10))
      DO 11 I=2,7
      EMB(I)=EM(I)
   11 CONTINUE
C STORE MASS OF BODIES 1 AND 2, AND RESET THE MASS CHANGE
C INDICATOR, MALTER, TO ZERO
   10 DO 12 I=1,2
      OLDEMB(I)=EMB(I)
   12 CONTINUE
      MALTER=0
C RESET INDEX AND H0 FOR SUBROUTINE DRIVE
      INDEX=1
      H0=1.0D-10
C INCREASE THE TIME STEP FOR NEXT OUTPUT AND CHECK TOTAL TIME
   30 TOUT=TOUT+TSTEP
      IF(TOUT.LE.TTL) GO TO 13
      WRITE(6,202)
  202 FORMAT(//' TOTAL TIME ALLOWED EXCEEDS',//)
      STOP
```

```
C CALL SUBROUTINE DRIVE OF EPISODE TO SOLVE THE EQUATIONS OF MOTION
C OF THE FORM DY/DT=F(Y,T); ARGUMENTS OF DRIVE ARE;
C         N= THE NUMBER OF EQUATIONS
C        T0= THE INITIAL VALUE OF T,AND IS USED FOR INPUT ONLY ON THE FIRST
C            CALL, ON OUTPUT, T0 IS THE OUTPUT VALUE OF T
C        H0= THE STEP SIZE H, H0 IS USED ON INPUT FOR THE STEP SIZE TO BE
C            ATTEMPTED FOR THE FIRST STEP, ON THE FIRST CALL
C        Y0= A VECTOR OF LENGTH N FOR THE DEPENDENT VARIABLE Y
C      TOUT= THE NEXT OUTPUT VALUE OF T
C       EPS= THE LOCAL ERROR TOLERANCE PARAMETER
C    IERROR= THE ERROR CONTROL INDICATOR
C        MF= THE METHOD FLAG
C     INDEX= AN INTEGER FLAG USED FOR INPUT AND OUTPUT
   13 CALL DRIVE(N,T0,H0,Y0,TOUT,EPS,IERROR,MF,INDEX)
C EXIT IF ERROR
      IF(INDEX.EQ.0) GO TO 14
      WRITE(6,203) INDEX
  203 FORMAT(//' ERROR RETURN WITH INDEX=',I3//)
      STOP
C SWITCH 'ON' OR 'OFF' OF LOGICAL FUNCTIONS EL(I) ACCORDING TO CRITERIA
   14 DX21=Y0(9)-Y0(8)
      DY21=Y0(2)-Y0(1)
      DX32=Y0(10)-Y0(9)
      DX72=Y0(14)-Y0(9)
      DO 15 I=1,24
      EL(I)=0
   15 CONTINUE
      IF(DX21-0.7292D0.GE.0.0D0) EL(1)=1
      IF(DX72-5.16667D0.LT.0.0D0) GO TO 16
      EL(2)=1
      KEL2=KEL2+1
   16 IF(DX21-0.91667D0.GE.0.0D0) EL(3)=1
      IF(DY21.GE.0.0D0) EL(4)=1
      EL(7)=EL(3)
      EL(8)=EL(4)
      IF(DX21-0.99167D0.GE.0.0D0) EL(9)=1
      EL(10)=EL(8)
      EL(11)=1-EL(9)
      EL(12)=1-EL(8)
      EL(13)=1-EL(3)
      EL(14)=EL(12)
      IF(DX21-0.99899D0.GE.0.0D0) EL(15)=1
      IF(-DX21+0.66667D0.GE.0.0D0) EL(18)=1
      EL(19)=EL(13)
      IF(-DY21.GE.0.0D0) EL(20)=1
      IF(-DX21+0.79167D0.GE.0.0D0) EL(21)=1
      IF(DX32-1.01667D0.GE.0.0D0) EL(22)=1
      IF(-DX32+0.98333D0.GE.0.0D0) EL(23)=1
      EL(24)=EL(2)
C CHECK END OF A CYCLE OF MOTION OF THE MECHANISM
      IF(EL(15).NE.1.OR.ELG(1).NE.1) GO TO 17
C COMPUTE AND OUTPUT RESULTS
      DX2X1=(Y0(9)-Y0(8)-1.0D0)*10.0D0
      DX3X2=(Y0(10)-Y0(9)-1.0D0)*100.0D0
```

```
      DX7X2=(Y0(14)-Y0(9)-5,0D0)*10,0D0
      DV7V1=(Y0(7)-Y0(1))*0,1D0
      DX6X1=(Y0(13)-Y0(8)-5,0D0)*10,0D0
      X1M1=(Y0(8)-1,0D0)*10,0D0
      WRITE(6,204) T0,DX2X1,DX3X2,DX7X2,DV7V1,DX6X1,X1M1
  204 FORMAT(' *END OF A CYCLE*',D8,3,6D16,5)
      STOP
C COMPUTE LOGICAL FUNCTION GROUPS ELG(I)
   17 ELG(1)=EL(1)*EL(4)*(1-EL(15))
      ELG(3)=EL(3)*ELG(1)
      ELG(4)=EL(19)*EL(20)
      ELG(5)=ELG(2)
      ELG(6)=EL(7)*EL(8)-EL(9)*EL(10)+EL(11)*EL(12)-EL(13)*EL(14)
      ELG(7)=1-ELG(1)
      ELG(8)=EL(18)
      ELG(9)=EL(19)
      ELG(10)=EL(20)*(EL(19)-EL(21))
      ELG(11)=ELG(1)
      ELG(12)=ELG(6)
      ELG(13)=EL(22)
      ELG(14)=EL(23)
      ELG(15)=EL(24)*ELG(7)
      ELG(16)=EL(15)
C COMPUTE THE MASS OF BODIES 1 AND 2
      EMB(1)=EM(1)+20,0D0*(EM(9)+EM(10))-ELG(1)*(EM(9)+EM(10))+(1,0D0
     *-ELG(7))*EM(7)-ELG(2)*EM(10)+ELG(3)*(EM(8)+EM(9)+EM(10))-ELG(4)*
     *(EM(8)+EM(9))
      EMB(2)=EM(2)+ELG(9)*EM(8)+ELG(10)*EM(9)+(ELG(11)-ELG(3))*(EM(9)
     *+EM(10))
C SPECIAL TREATMENT FOR THE IMPULSIVE FORCE FGAS ; IF KEL2=1 COMPUTE
C NEW INITIAL VALUES THROUGH JUMP CONDITIONS AND LOCK ON ELG(2);
C OTHERWISE, CONTINUE COMPUTATION
      IF(KEL2,NE,1) GO TO 50
      Y0(1)=(OLDEMB(1)*Y0(1)-1,2D0)/EMB(1)
      Y0(2)=(OLDEMB(2)*Y0(2)-0,9D0*1,2D0)/EMB(2)
      ELG(2)=1
      MALTER=1
      GO TO 70
C CHECK MASS CHANGE OF BODIES 1 AND 2; IF YES, COMPUTE NEW INITIAL
C VALUES THROUGH JUMP CONDITIONS; OTHERWISE, CONTINUE
   50 DO 20 I=1,2
      IF(DABS(EMB(I)-OLDEMB(I)),LE,1,0D-15) GO TO 20
      MALTER=1
      Y0(I)=OLDEMB(I)*Y0(I)/EMB(I)
   20 CONTINUE
C COMPUTE AND OUTPUT RESULTS
   70 DX2X1=(Y0(9)-Y0(8)-1,0D0)*10,0D0
      DX3X2=(Y0(10)-Y0(9)-1,0D0)*100,0D0
      DX7X2=(Y0(14)-Y0(9)-5,0D0)*10,0D0
      DV7V1=(Y0(7)-Y0(1))*0,1D0
      DX6X1=(Y0(13)-Y0(8)-5,0D0)*10,0D0
      X1M1=(Y0(8)-1,0D0)*10,0D0
      IF(MALTER,EQ,1) GO TO 60
      WRITE(6,205) T0,DX2X1,DX3X2,DX7X2,DV7V1,DX6X1,X1M1
```

```
  205 FORMAT(15X,D10.3,6D16.5)
      GO TO 30
   60 WRITE(6,206) T0,DX2X1,DX3X2,DX7X2,DV7V1,DX6X1,X1M1
  206 FORMAT(' *MASS CHANGE* ',D10.3,6D16.5)
      GO TO 40
      END
C
C SUBROUTINE DIFFUN IS CALLED BY DRIVE TO COMPUTE YDOT=F(Y,T) OF LENGTH
C N FOR GIVEN VALUES OF T AND Y OF LENGTH N
      SUBROUTINE DIFFUN(N,T,Y,YDOT)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      INTEGER FL,ELG
      DIMENSION Y(N),YDOT(N)
      DIMENSION F(14),EL(24),ELG(16),FM(10),EMB(10),OLDEMB(10)
      COMMON EL,ELG,KEL2,EM,EMB
C COMPUTE FORCES IN THE FIRST 7 EQUATIONS OF MOTION
      FMOUNT=-300.0D0*(Y(8)-1.0D0)-9.53D0*Y(1)
      F12BAR=-10000.0D0*(Y(8)-Y(9)+0.66667D0)-0.13D0*(Y(1)-Y(2))
      F12BB=-20000.0D0*(Y(8)-Y(9)+0.99967648D0)-25.0D0*(Y(1)-Y(2))
      F16=-76.8D0*(Y(8)-Y(13)+5.08425D0)-(0.0217D0*DSQRT(2.0D0*EM(5)*76.
     *8D0))*(Y(1)-Y(6))
      F17=-15.0D0*(Y(8)-Y(14)+6.5D0)-(0.98D0*DSQRT(2.0D0*FM(7)*15.0D0))*
     *(Y(1)-Y(7))
      F23=-20000.0D0*(Y(9)-Y(10)+1.016667D0)-2.0D0*(Y(2)-Y(3))
      F23BAR=-20000.0D0*(Y(9)-Y(10)+0.98333D0)-2.0D0*(Y(2)-Y(3))
      F24=-76.8D0*(Y(9)-Y(11)+1.91575D0)-(0.0217D0*DSQRT(2.0D0*FM(5)*76.
     *8D0))*(Y(2)-Y(4))
      F27=-20000.0D0*(Y(9)-Y(14)+5.166667D0)-36.0D0*(Y(2)-Y(7))
      F45=-76.8D0*(Y(11)-Y(12)+0.91575D0)-(0.0217D0*DSQRT(2.0D0*EM(5)*76
     *.8D0))*(Y(4)-Y(5))
      F56=-76.8D0*(Y(12)-Y(13)+0.91575D0)-(0.0217D0*DSQRT(2.0D0*FM(5)*76
     *.8D0))*(Y(5)-Y(6))
      F21BAR=-F12BAR
      F21BB=-F12BB
      F32=-F23
      F32BAR=-F23BAR
      F42=-F24
      F54=-F45
      F61=-F16
      F65=-F56
      F71=-F17
      F72=-F27
C COMPUTE THE RIGHT SIDES OF THE FIRST 7 EQUATIONS OF MOTION
      F(1)=(F16+FMOUNT+ELG(7)*F17+ELG(8)*F12BAR+ELG(16)*F12BB)/EMB(1)
      F(2)=(F24+ELG(13)*F23+ELG(14)*F23BAR+ELG(15)*F27+ELG(8)*F21BAR+ELG
     *(16)*F21BB)/EMB(2)
      F(3)=(ELG(13)*F32+ELG(14)*F32BAR)/EMB(3)
      F(4)=(F42+F45)/EMB(4)
      F(5)=(F54+F56)/EMB(5)
      F(6)=(F65+F61)/EMB(6)
      F(7)=(ELG(7)*F71+ELG(15)*F72)/EMB(7)
C SPECIAL TREATMENT FOR FCAM ,THE CAM FORCE
      IF(ELG(6).EQ.0) GO TO 30
      PI=3.141592653D0
```

```
      EPSLOC=0.075D0
      C1=PI/EPSLOC
      DX1X2=Y(8)-Y(9)+0.99167D0
      ELP=0.5D0*C1*DSIN(C1*DX1X2)
      ELPP=0.5D0*C1*C1*DCOS(C1*DX1X2)
      GAMLOC=0.3927D0
      ENRT=10.5D-7*FLG(1)+1.0D-7
      A=GAMLOC*GAMLOC*ENRT*ELP*ELP
      B=-GAMLOC*GAMLOC*ENRT*ELP*ELPP*(Y(2)-Y(1))**2
      A11=EMB(1)+A
      A12=-A
      A21=-A
      A22=EMB(2)+A
      B1=F(1)*EMB(1)+B
      B2=F(2)*EMB(2)-B
      D=A11*A22-A21*A12
      F(1)=(A22*B1-A12*B2)/D
      F(2)=(A11*B2-A21*B1)/D
C COMPUTE THE RIGHT SIDES OF THE LAST 7 EQUATIONS OF MOTION
   30 DO 10 I=1,7
      F(I+7)=Y(I)
   10 CONTINUE
C STORE THE RIGHT SIDES OF THE EQUATIONS OF MOTION IN VECTOR YDOT
      DO 20 I=1,14
      YDOT(I)=F(I)
   20 CONTINUE
      RETURN
      END
C
C THIS DUMMY VERSION OF PEDERV IS REQUIRED BY THE ODE SOLVER EPISODE
      SUBROUTINE PEDERV(N,T,Y,PD,N0)
      RETURN
      END
```

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
2026

2. GOVT ACCESSION NO.
AD-A083 809

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
COMPUTER CALCULATION OF MECHANISMS INVOLVING INTERMITTENT MOTIONS

5. TYPE OF REPORT & PERIOD COVERED
Summary Report - no specific reporting period

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
B. Noble and H. S. Hung

8. CONTRACT OR GRANT NUMBER(s)
DAAG29-75-C-0024

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Mathematics Research Center, University of Wisconsin
610 Walnut Street
Madison, Wisconsin 53706

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Work Unit #3 - Applications of Mathematics

11. CONTROLLING OFFICE NAME AND ADDRESS
U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, North Carolina 27709

12. REPORT DATE
December 1979

13. NUMBER OF PAGES
38

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Mechanical systems
Intermittent motion
Heaviside step-functions
Logical functions
Jump conditions
Equations of motion
Dynamical analysis
Computer program

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper deals with a simple computational approach to the analysis of dynamical systems involving intermittent motion in which the velocities involved can be discontinuous due to impulsive forces, impact, mass capture, and mass release. The sequence of these events may not be known ahead of time, and may in fact be one of the things we wish the computer to determine.

next page

20. ABSTRACT - Cont'd.

The dynamical equations are formulated using a logical function method due to P. Ehle. The resulting system of ordinary differential equations with discontinuous coefficients is integrated using a standard computer code in regions where the coefficients are continuous. When discontinuities occur, jump conditions across the discontinuity are used to express the new velocities in terms of the old, and the ordinary differential equation solver is simply restarted with new initial conditions.

To illustrate the simplicity of the approach, the method is applied to a dynamical system of ten masses considered by Ehle. The computer code and numerical results are included.